JOHANNES KEPLER
UNIVERSITÄT LINZ

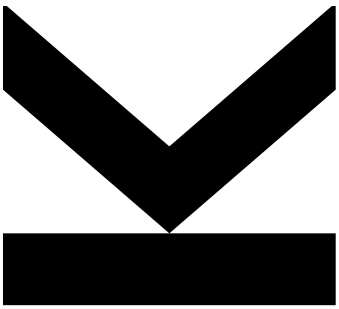
Submitted by
Michael Reinegger
k12102754
Julian Weichinger
k12110801

Submitted at
Institute for Application-Oriented Knowledge Processing

Lecturer
A.Univ.-Prof., DI Dr. Josef Küng

November 2026

An Overview on Zero-Knowledge Proofs and Real-World Applications



Seminar Thesis

in the Seminar in Networks and Security: 351034 (2025W)



JOHANNES KEPLER UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
www.jku.at
DVR 0093696

Abstract

Zero-knowledge proofs are basically a way to prove you know a secret without having to reveal the secret itself. This idea has been around since 1985, but it's recently gone from being just mathematical theory to a real-world implementations actually used in modern information systems.

In this paper, we are going to walk through the main foundations of how Zero-Knowledge Proofs (ZKPs) work. Starting with the basics of interactive and non-interactive proofs as well as the three major formal properties: completeness, soundness, and zero-knowledge.

Later on, we'll look at some Use Cases and cover how ZKPs are used in modern systems for privacy and authentication, specifically looking at how Zcash uses zk-SNARKs on large scale.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Structure of this Work 1

2 Definition and Properties **2**
2.1 What is a proof? 2
2.2 Properties 3

3 ZKP **5**
3.1 Definition 5
3.2 How ZKPs Work/Basic Examples 5
3.3 Different Types of ZKP 7

4 Related Work and Applications **9**
4.1 ZKPs for Transactions 9
4.2 ZKPs for Privacy 9
4.3 ZKPs for Identification 10

5 Conclusion **12**

A Acronyms **14**

1 Introduction

Proofs, in computer science and cryptography, are essential for showing that a statement is valid. However, this often involves the usage of secret or confidential information, which might be considered a compromise of security. To solve this, the concept of ZKP was introduced in the 1980s by the researchers Shafi Goldwasser, Silvio Micali, and Charles Rackoff [1]. They proposed a new way to look at proofs, where it is possible to demonstrate that a theorem or claim is true without giving away any hint as to why it is true [1]

The basic idea of a ZKP involves two parties: a "prover" and a "verifier." The prover wants to convince the verifier that they know a specific secret or that a statement is correct. The challenge is to do this without revealing the secret itself to the verifier or any eavesdropper watching the communication [1]. This is defined as the verifier, after talking to the prover, is only able to compute what he was able to compute before the communication, meaning no new information can be gained [2].

Today, this technology is moving from theory to real-world use, especially in areas like Blockchain technology and secure authentication [1], where anonymity or the security of private information are essential for a convenient and secure use. This can be seen with applications like Zcash, Ethereum, and even for private messaging with tools like Semaphore [3].

1.1 Motivation

The main idea of using ZKP is to be able to establish trust in a system without compromising the privacy or security of the system. In many standard cryptographic protocols, a user must send private data to a server to prove who they are. ZKPs are used because they allow a party to prove a statement is true without disclosing the underlying data or secrecy to the other party [1].
This property is useful for many different applications. For example, in authentication systems, a user can prove they know a password without actually sending the password over the network, which makes it harder for attackers to steal credentials. Another significant motivation is found in blockchain technology. Public blockchains are very open as they show all transaction data to everyone. By using ZKPs, systems like Zcash can validate transactions while keeping the sender, recipient, and amount private [4].

1.2 Structure of this Work

This work is structured as follows: Section 2 introduces the formal definition and properties of ZKP. In the first part of Section 3 popular examples of Zero-Knowledge proofs are illustrated and explained, moving on to different types of ZKP and selected protocols. Section 4 discusses related work and various applications of ZKP, including their use for privacy preservation. Finally, Section 5 provides concluding remarks.

2 Definition and Properties

2.1 What is a proof?

In computer science and mathematics, a proof is a formal method for demonstrating that a particular statement or claim is true within a given logical framework. At its core, a proof consists of a sequence of logically valid steps that follow from a set of axioms, definitions, and previously established results. By connecting these steps in an argument, the proof confirms the correctness of a theorem or proposition without any reasonable doubt. [5, p. 2]
While classical proofs consist of a sequence of logical arguments, interactive proofs (IP) take this approach and expand it by adding a communication between two parties: a prover and a verifier. The verifier engages in a dialogue with a prover, issuing random challenges to the prover and evaluating the responses. The use of a randomly chosen challenge introduces a small probability of error: an incorrect statement might be accepted with a negligible, non-zero likelihood. [5, p. 2] Despite this, IP maintain strong guarantees. When the prover is honest and the statement is true, the verifier will accept it with very high probability. When the statement is false, a dishonest prover can only convince the verifier with a small probability. The strength of IP is closely tied to two foundational results in complexity theory.

First, the notation IP = PSPACE denotes that IP-systems are capable of deciding exactly those problems that can be solved using polynomial space. Since PSPACE is widely believed to be strictly larger than NP, this result suggests that IPs are significantly more expressive than classical nondeterministic Polynomial Time (NP)-style proofs, which correspond to the special case where the verifier performs no randomization. Second, the Probabilistically Checkable Proof (PCP) theorem demonstrates that any statement that can be verified through a conventional proof can instead be checked by inspecting only a constant number of randomly chosen bits from a specially encoded proof. This insight reshaped our understanding of proof verification and forms the basis for many modern proof systems. [5, p. 2]

2.2 Properties

IP systems are characterized by two fundamental security properties. The first one is computational *soundness*, which states that no malicious prover can convince the verifier to accept a false statement as true, except with negligibly small probability. Complementing this is the property of *completeness*, which requires that any valid statement, when presented by an honest prover, is always accepted by the verifier. A third property that specifically applies to ZKPs is *Zero-Knowledge*, which states that the protocol must not reveal any information beyond the fact that the statement being proven is true, thereby keeping any secret data involved private. These properties can be formalized (cf. [1] [5]):

1. **Completeness:** If a statement is true, an honest prover holding a valid witness w can convince the honest verifier of its validity with overwhelming probability:

$$\forall x \in L, w \in R(x): \quad \Pr\big[\langle P(w), V\rangle(x) = 1\big] \geq 1 - \mathsf{negl}(|x|)$$

2. **Soundness:** If a statement is false, no malicious prover P^* can convince the honest verifier of the validity of this statement, except with negligible probability:

$$\forall P^*, \forall x \notin L: \quad \Pr\big[\langle P^*, V\rangle(x) = 1\big] \leq \mathsf{negl}(|x|)$$

3. **Zero-Knowledge:** If a statement is true, a verifier learns nothing about the underlying witness w beyond the truth of the statement. Formally, for every Probabilistic Polynomial-Time (PPT) malicious verifier V^*, there exists a PPT simulator S such that for all $x \in L$ and $w \in R(x)$, the simulated transcript is computationally indistinguishable from the real verifier's view:

$$\big\{\mathrm{View}_{V^*}\big\langle P(w),\, V^*\big\rangle(x)\big\}_{x\in L} \;\approx_c\; \big\{S(x)\big\}_{x\in L}$$

To illustrate the properties, we came up with an explanatory, fictional story:

We define two participants, Alice (the prover) and Bob (the verifier). Alice claims that she knows the correct combination to a digital safe. Bob cannot see the combination himself, but he wants Alice to convince him that she actually knows it, without learning anything about the combination.

To test Alice's claim, Bob uses a verification procedure that allows him to check whether Alice can open the safe on demand. Important to note is that, Alice is never asked to reveal the combination itself. She only has to demonstrate that she can successfully use it.

Completeness: If Alice truly knows the correct combination and follows the protocol honestly, she will always be able to open the safe when Bob asks her to do so. As a result, Bob will accept Alice's proof every time. This shows that the protocol satisfies completeness: a true statement can always be verified by an honest prover.

Soundness: If Alice does not know the combination, she can only guess. In this case, her chance of successfully opening the safe in a single attempt is very low. By repeating the verification multiple times, the probability that Alice can consistently succeed without knowing the combination becomes negligible. This means that a dishonest prover can't convincingly trick Bob.

Zero Knowledge: Throughout the entire process, Bob never learns anything about the combination itself. He only observes whether the safe opens or not. Even after many successful verifications, Bob gains no additional information about the underlying secret, beyond the fact that Alice must know it.
Other properties mentioned for making common ZKPs more applicable for real-world use, making them efficient, are defined (cf. [5]):

1. **Succinctness:** Proofs should be efficiently verifiable and relatively short compared to the statement, which has to be proven.
2. **Non-Interactivity:** Proofs can be sent and verified without the need to have a running conversation between the two parties.

3 ZKP

3.1 Definition

Zero-knowledge proofs (ZKP), first introduced in 1985, are cryptographic techniques that allow one party (the prover) to convince another party (the verifier) that a statement is true without revealing any additional information about the underlying data or claim.

The basic idea of ZKPs is that a prover has to convince a verifier that he knows a secret or that a statement is true without revealing the secret itself or why a statement is true to the verifier. One important point to note is that a ZKP does not guarantee correctness with absolute certainty in a single execution. Instead, there is always a minimal probability that a dishonest prover could convince the verifier by chance. To achieve a sufficiently high level of confidence, the protocol can be executed repeatedly, which reduces this probability to a negligible level and increases the verifiers trust in the correctness of the provers claim. [6]

In literature, two different types of ZKPs are described: Interactive and Non-Interactive Proofs. These will be further discussed in section 3.3.

3.2 How ZKPs Work/Basic Examples

To illustrate the concept of ZKPs, we present two intuitive examples of interactive Zero-Knowledge Proofs that are commonly used in literature.

Red-Green Ball

To illustrate the idea of a ZKP in an intuitive way, consider an example involving color perception (cf. [7]). Assume that the prover possesses two balls of different colors, one red and one green, while the verifier is color-blind and therefore unable to distinguish between them. The prover's goal is to convince the verifier that the two balls are indeed different in color (statement), without revealing which ball has which color (secret).

At the beginning, the verifier takes one ball in each hand and shows them to the prover. Since the prover can distinguish the colors, they can clearly identify which ball is red and which is green. The verifier then hides both hands behind their back, while keeping track of which ball is held in which hand. Afterwards, the verifier either swaps the balls between their hands or leaves them in the same position, and then presents them to the prover again. The prover is asked to determine whether a swap has occurred, without disclosing any information about the actual colors of the balls.

Because the prover can distinguish between red and green, they are able to correctly identify whether the balls were exchanged and communicate this decision to the verifier. From the verifier's perspective, a single correct answer could still be the

result of random guessing, as there is a 50% chance of success. However, by repeating this procedure multiple times, the probability that the prover consistently answers correctly by chance becomes negligibly small. As a result, the verifier is convinced that the two balls must indeed be of different colors, even though no information about which ball is red or green is ever revealed.

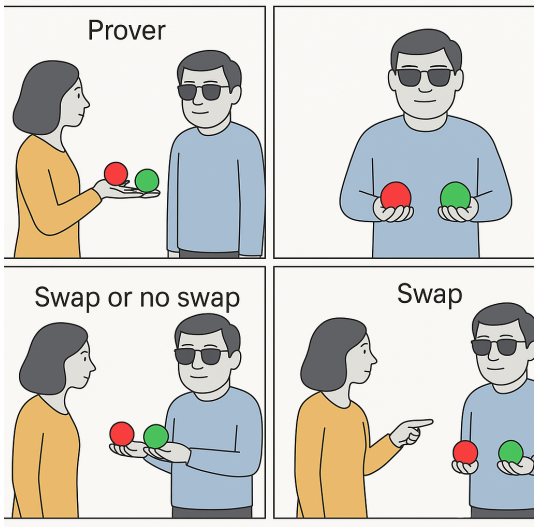


Figure 3.1: Red Green Ball Example (generated with ChatGPT 5.2)

Alibaba Cave

Another example of a ZKP is the Alibaba cave (cf. [7]). It involves two participants, called Peggy (the prover) and Victor (the verifier), who discover a ring-shaped cave. The cave has a single entrance and a magic door located on the opposite side, which can only be opened using a secret word. Peggy claims to know this secret word but does not want to reveal it to Victor. Victor has to be convinced that Peggy actually knows the secret.

Firstly, Peggy and Victor agree on naming the two paths inside the cave path A and path B. Peggy then enters the cave while Victor stays outside, so he cannot see which path Peggy chooses. Once Peggy has chosen a path and walked to the magic door, Victor enters the cave entrance and randomly calls out one of the two paths, A or B, asking Peggy to return via the specified path.

If Peggy truly knows the secret word, she can always come back the correct path Victor requested. She can open the magic door if necessary and return through either path. If Peggy does not know the secret word, she can only return via the path she initially took. In this case, she succeeds only if Victor requested the same path, which occurs with a probability of 50%. If she returns the wrong path, Victor can instantly expose that Peggy did not know the secret to opening the magic door.

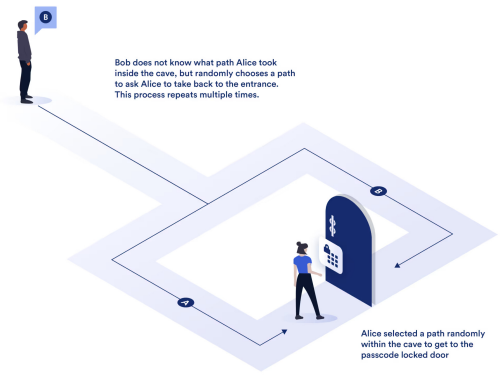


Figure 3.2: Alibaba Cave [8]

Since a single round does not provide strong evidence that Peggy knows the secret, Peggy and Victor repeat this procedure multiple times. With each successful repetition, Victor's confidence that Peggy knows the secret word increases. Victors confidence can never reach absolute certainty, as there always remains a small probability that Peggy was simply guessing correctly in every iteration.

3.3 Different Types of ZKP

ZKPs can be broadly classified into interactive and non-interactive approaches, depending on the communication pattern between the prover and the verifier (cf. [6]).

Interactive Zero-Knowledge Proofs

Interactive Zero-knowledge Proofs represent the classical form of ZKPs and rely on a sequence of exchanges between the prover and the verifier. At first the prover sends a commit to the verifier to start the process. After the verifier received the commit, he issues a random challenge to the prover, who must respond with an according response. Through multiple iterations of this interaction, the verifier can achieve a high level of confidence in the truthfulness of the claim. Two key characteristics define interactive ZKPs. First, numerous interactions are required before the verifier becomes sufficiently convinced. Second, the resulting proof is not transferable to third parties, as external observers cannot be certain that the verifier's challenges were not coordinated with the prover. Therefore, interactive ZKPs are primarily suited to scenarios involving a single verifier. Popular examples in literature for interactive ZKPs are the Schnorr identity and Chaum-Pedersen protocol.

Non-Interactive Zero-Knowledge Proofs

Non-interactive Zero-Knowledge Proofs eliminate the need for repeated exchanges between the prover and the verifier. Instead, the prover generates a proof that can be verified in a single step, without further communication. This property allows such proofs to be transferred and verified by multiple third parties, making them particularly suitable for distributed and decentralized systems.

The transition from interactive to non-interactive proofs is typically achieved by replacing the verifier's random challenges with a common reference value, such as a publicly known random string. Using this shared reference, the prover simulates the interactive process and constructs a proof that the verifier can subsequently validate by independently reproducing the verification steps.
Non-interactive ZKPs generally require greater computational resources than interactive protocols. Nevertheless, their scalability and transferability make them advantageous in applications where proofs must be verified by large groups of participants. Popular examples for non-interactive ZKPs are zk-SNARKS and zk-STARKS. The figure below shows the protocol flows for the different types of ZKPs:

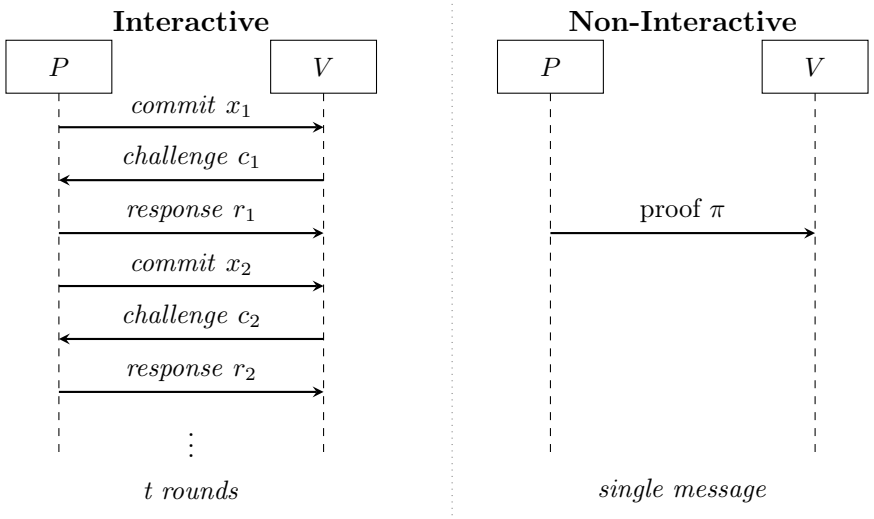


Figure 3.3: Interaction between Verifier and Prover: Interactive vs. Non-Interactive Zero-Knowledge Proofs

4 Related Work and Applications

4.1 ZKPs for Transactions

Zcash is a cryptocurrency that was launched in 2016 and is built on the Bitcoin protocol, but with added privacy features. Unlike Bitcoin, where all transaction data is publicly visible on the blockchain, Zcash allows users to make so-called shielded transactions, where the sender, receiver, and amount are all kept private [4].
To achieve this, Zcash uses a specific type of non-interactive ZKP called zk-SNARK, which stands for Zero-Knowledge Succinct Non-Interactive Argument of Knowledge. A zk-SNARK allows a user to prove that a transaction is valid, meaning that the sender actually owns the funds they are trying to spend, without revealing any of the transaction details to the network [4].
In practice, not all Zcash transactions are shielded. Kappos et al. [9] analyzed the Zcash blockchain and found that the majority of transactions at the time were still transparent, meaning they worked essentially like Bitcoin. This limits the actual privacy guarantees in practice, because users who do use shielded transactions form a relatively small anonymity set. The study shows that just having a privacy feature available is not enough, the feature also has to be widely used to actually protect individual users.
Still, Zcash is one of the most prominent real-world examples of ZKPs being deployed at scale, and it demonstrates that the computational overhead of zk-SNARKs is manageable enough for a live production system.

4.2 ZKPs for Privacy

One of the main reasons why ZKPs are gaining attention is the growing need to protect personal data online. According to Jedlicka and Grant [10], around 17.2 billion user accounts have been affected by data breaches since 2004. The problem is that most online services require users to share sensitive information like passwords, banking details, or social security numbers just to use them. Once this data is handed over, users have little to no control over what happens to it. Laws like the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) try to address this by giving users certain rights, such as knowing what data is collected and being able to request its deletion. However, these laws still assume that data has to be sent to a service provider first, which means it can still be stolen or misused.
ZKPs could change this by allowing users to prove things about themselves, for example that they are old enough, that they have a valid identity, or that their credit score meets a threshold, without actually sending the data to the other party. Several approaches have been proposed to make this work in practice. One example is a blockchain-based identity management system that uses Ethereum smart contracts together with ZKPs, so users can verify their identity to a service provider without revealing any personal details [10]. Another example is a privacy-preserving credit score system that combines non-interactive ZKPs with homomorphic encryption, meaning a credit score can be calculated and checked without any identifiable

information ever leaving the user's device. For web authentication, researchers have also proposed ZKP-based protocols that replace traditional username and password logins entirely by using elliptic curve cryptography to verify both the server and the user, which would make phishing attacks and password theft largely ineffective [10]. Blockchain-based solutions offer transparency and tamper-proof records, but storing large amounts of data on a blockchain is still expensive and does not scale well [10]. Generating and verifying ZKPs also requires a certain amount of computing power, which can be a problem for devices with limited resources like smart cards or Internet of Things (IoT) sensors. Managing cryptographic keys without relying on a trusted third party also adds complexity. Still, the various proposals show that ZKPs are a realistic approach toward a system where users keep control over their own data and only share what is strictly needed for a specific purpose.

4.3 ZKPs for Identification

One of the earliest and most impactful applications of ZKPs is in the domain of identity verification. Traditional identification mechanisms, such as passwords, PIN numbers, or physical ID cards, rely on revealing a fixed secret to prove one's identity. This fundamental weakness allows a sophisticated adversary, potentially cooperating with a dishonest verifier, to record or copy the secret and later impersonate the legitimate party. Feige, Fiat, and Shamir [11] addressed this problem by proposing an identification scheme based on zero-knowledge proofs of knowledge, in which a party proves their identity by demonstrating knowledge of a secret rather than by disclosing it. Their scheme is provably secure under the assumption that integer factoring is computationally difficult, and its practical implementations require roughly two orders of magnitude fewer modular multiplications than Rivest,Shamir,Adleman Algorithm (RSA)-based alternatives, making it suitable even for resource-constrained devices such as smart cards.

The scheme operates as follows: A trusted center publishes a modulus n, which is the product of two large primes. Each user selects k random secret values $S_1, \ldots, S_k$ and derives corresponding public keys $I_j = \pm 1/S_j^2 \pmod{n}$. To prove their identity, the prover and verifier engage in t rounds of a challenge-response protocol. In each round, the prover commits to a random value, the verifier issues a random binary challenge vector, and the prover responds with a value that the verifier can check against the public keys. Crucially, the verifier never learns anything about the secret values S_j themselves. The protocol satisfies the zero-knowledge property because a simulator can reproduce the verifier's view without access to the secrets, and it satisfies soundness because any party that consistently passes verification must possess knowledge of the S_j, which can be formally extracted by a polynomial-time machine [11].

An notable extension of this eliminates the need for a public key directory entirely. In this variant, the trusted center derives each user's public keys pseudo-randomly from a string I containing the user's name, address, and other identifying information, computes the corresponding secret square roots, and stores them on a smart card. When a user wishes to prove their identity, the verifier can reconstruct the public

keys directly from the claimed identity, removing the need to consult an external directory. This approach offers provable identification for all participants: provers cannot deceive verifiers, verifiers cannot later impersonate provers, and even coalitions of provers and verifiers cannot forge new identities or recover the factorization of n [11].

5 Conclusion

In this paper, we took a closer look at ZKPs, starting from the basic concept of interactive proofs and the three core properties that define them: completeness, soundness, and zero-knowledge. These properties were first formally defined and then illustrated with intuitive examples to help make the complex concept more accessible.

ZKPs solve a genuinely difficult problem: how to prove something is true without giving away why it is true. This makes them useful in any situation where trust and privacy are both required at the same time.

How ZKPs do this is explained in the core of the paper, where we give the reader a deeper understanding of Zero-Knowledge Proofs, illustrating easy but very important examples of ZKPs like the Alibaba Cave. Another very important chapter is the differentiation between interactive and non-interactive ZKPs, where key differences where summarized and explained.

Future work in this area is mostly focused on making ZKPs faster and cheaper to compute and on making non-interactive variants like zk-SNARKs easier to deploy without requiring a trusted setup. By reducing these practical limitations, it is likely for ZKPs to appear more often in new systems, especially in authentication and privacy-preserving data verification.

References

[1] J. Hasan, "Overview and applications of zero knowledge proof (zkp)," *International Journal of Computer Science and Network*, vol. 8, no. 5, pp. 2277–5420, 2019.

[2] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, p. 1–32, Dec. 1994. [Online]. Available: http://dx.doi.org/10.1007/BF00195207

[3] matter labs, "awesome-zero-knowledge-proofs," https://github.com/matter-labs/awesome-zero-knowledge-proofs, 2018, accessed: 2025-12-14.

[4] D. Hopwood, S. Bowe, T. Hornby, N. Wilcox *et al.*, "Zcash protocol specification," *GitHub: San Francisco, CA, USA*, vol. 4, no. 220, p. 32, 2016.

[5] J. Ernstberger, S. Chaliasos, L. Zhou, P. Jovanovic, and A. Gervais, "Do you need a zero knowledge proof?" Cryptology ePrint Archive, Paper 2024/050, 2024. [Online]. Available: https://eprint.iacr.org/2024/050

[6] I. Aad, "Zero-knowledge proof," *Trends in Data Protection and Encryption Technologies*, pp. 25–30, 2023.

[7] Zero-knowledge proofs. https://aimultiple.com/zero-knowledge-proofs. Accessed: 2025-12-09.

[8] Chainlink, "Zero-knowledge proof (zkp) - explained: Chainlink," Jul 2024, accessed: 2025-12-14. [Online]. Available: https://chain.link/education/zero-knowledge-proof-zkp

[9] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in zcash," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 463–477.

[10] J. Jedlicka and E. S. Grant, "Data privacy through zero-knowledge proofs," in *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*. IEEE, Dec. 2022.

[11] U. Fiege, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity," in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '87. New York, NY, USA: Association for Computing Machinery, 1987, p. 210–217. [Online]. Available: https://doi.org/10.1145/28395.28419

A Acronyms

ZKP Zero-Knowledge Proof

IP interactive proofs

PCP Probabilistically Checkable Proof

NP nondeterministic Polynomial Time

PPT Probabilistic Polynomial-Time

CCPA California Consumer Privacy Act

GDPR General Data Protection Regulation

RSA Rivest,Shamir,Adleman Algorithm

IoT Internet of Things